REMARKS

Claims 1 and 11 have been amended. No claims have been cancelled or added. Hence, Claims 1 - 20 are pending in the Application.

As a preliminary matter, receipt of the Notice of Draftsperson's Patent Drawing Review is acknowledged. Applicant has furnished corrected drawings that incorporate corrections that fix the informalities noted in the review. No new matter has been added.

SUMMARY OF REJECTIONS 1

Claims 1 – 20 are rejected under 35 U.S.C. 103 as being unpatentable over U.S. Patent No. 5,862,331, issued to Robert G. Herriot on Jan. 19, 1999, herein *Herriot*, in view of U.S. Patent No. 5,835,766, issued to Hiroiku Iba, et al., on Nov. 10, 1998, herein *Iba*. These rejections are respectfully traversed.

Claims 1 and 11, recite:

registering in a name service participant data that identifies a plurality of participants that are participating in said distributed operation;

wherein said distributed operation is a unit of work involving said plurality of participants;

wherein said name service registers information received from clients and provides said information to clients that request the information; and

causing a node that requires information about participants in said distributed operation to retrieve said participant data from said name service.

The rejections appear to be based on an ambiguity or a misunderstanding of the terms "operation" and "name service". In response, claims 1 and 11 have been amended to clarify of the terms.

For example, the office action as apparently analogized a distributed operation to a "distributed computing environment". The specification makes clear what a distributed operation

is, defining it to be "an operation in which processes on multiple nodes participate in the accomplishment of the operation." Thus, an operation is something that can be performed and accomplished, like a task or a "unit of work." A unit of work can be accomplished; a distributed computing environment cannot.

Applicant admits that *Herriot* teaches that a name service identifies components of a distributed computing environment that are capable of performing work, including clients and servers. In teaching this, *Herriot* suggests identifying entities that can participate in a distributed operation. However, *Herriot* does not disclose or suggest in any way that the participants involved in a particular unit of work are identified in any way. In fact, nothing in *Herriot* suggests that a particular task or instance of a service performed by servers or clients are tracked in any way. At best, *Herriot* teaches to identify the entities available to provide work, but does not teach to identify or track who participated in a particular unit of work.

Iba too also fails to disclose or suggest limitations of claims 1 and 11. For example, *Iba* fails to disclose or suggest in anyway the step of registering participant data in a name service. In *Iba*, local lock managers transmit registration requests to a global deadlock detector when a lock on a resource cannot be granted. The global deadlock detector stores the information and uses it to detect global deadlocks, as follows.

> The local lock manager LLM receives a request to acquire or release hardware resources to lock or unlock resources from an application program. For a resource lock request, if the resource can be occupied by the application program, processing after the processing of the local lock manager returns to the transaction manager. If the resource cannot be locked, occupation of the resource, i.e., the identifier of transaction waiting for locking, and the identifier of transaction currently occupying the resource are notified to the global deadlock detector GDD32, followed by the contents being registered with the WFG management

> table 33. When receiving the resource release request, the LLM notifies the GDD32 of the identifier of a transaction to be unlocked of the GDD32, followed by eliminating the registration of the transaction from the WFG management table 33.
>
> The global deadlock detector 32 receives a registration request or a deletion request of contents of the WFG graph from the local lock manager inside the respective resource managers. When receiving the request to register contents of the WFG, the global deadlock detector 32 registers the contents, and then it is determined whether or not there are deadlocks between global transactions. (col 9, ln 24 - 45)

As shown above, the global deadlock detector determines whether a global deadlock exists based on registration information in the WFG management table. Applicant has been unable to find any description in *Iba* that discloses or suggests that any other entity determines the existence of global deadlocks. There is therefore no need for the system of *Iba* to provide registration information to any other entity other than the global deadlock detector, or to share access to the table where the registration information is stored. In fact, a key distinction that *Iba* makes between its system and the prior art is that only the transaction manager with the global lock detector has a table for the registration information, as follows.

> Subsequently, an example to detect a deadlock generated so as to stretch over resource managers between global transactions will be explained with reference to FIG. 9 by comparing with the conventional example shown in FIG. 5. Comparing FIG. 9 with FIG. 5 results in finding that the following points are different between both of the FIGS., i.e., (1) the transaction manager 30 is provided with a global deadlock detector 32 and a WFG management table 33 for managing wait-for relation between global transactions; and (2) respective resource managers 31a and 31b are not provided with a WFG management table. (col 9 ln 59 – col 10 ln 2)

In teaching that only the global deadlock detector determines global deadlocks, and that the registration information the global deadlock detector uses to determine global deadlocks is not shared with other entities, *Iba* teaches way from the step of registering participant data with a name service, where it can be shared and accessed by multiple clients.

For the foregoing reasons, neither *Herriot* nor *Iba* disclose or suggest in any way the limitations of claims 1 and 11. Therefore, claims 1 and 11 are patentable. Reconsideration and allowance of claims 1 and 11 is respectfully requested.

Claims 2 – 10 and 12 – 20 are dependent claims, each of which depends (directly or indirectly) on one of the claims discussed above. Each of the dependant claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of the dependant claims introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case, a separate discussion of those limitations is not included at this time.

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: December 7, 2001

Marcel K. Bingham
Reg. No. 42,327

1600 Willow Street
San Jose, CA 95125
Telephone No.: (408) 414-1080 ext.206
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Box No Fee Amendment, Assistant Commissioner for Patents, Washington, DC 20231.

on Dec. 7, 2001 (Date) by Trudy Bagdon (Signature)

MARKED-UP VERSION OF CLAIMS

1. (Amended) A method of determining participants of a distributed operation in a distributed system, the method comprising the steps of:

registering in a name service participant data that identifies a plurality of participants that are participating in said distributed operation; **[and]**

wherein said distributed operation is a unit of work involving said plurality of participants;

wherein said name service registers information received from clients and provides said information to clients that request the information; and

causing a node that requires information about participants in said distributed operation to retrieve said participant data from said name service.

2. (No Change) The method of Claim 1, wherein the step of causing a node to retrieve said participant data includes causing said node to retrieve said participant data in response to said node performing deadlock detection.

3. (No Change) The method of Claim 1, wherein:

said distributed operation is a distributed transaction; and

the step of registering includes registering in a name service participant data that identifies which database servers of a plurality of database servers are participating in said distributed transaction.

4. (No Change) The method of Claim 1, further including the step of causing updates to said participant data to identify a new participant in said distributed operation.

5. (No Change) The method of Claim 4, wherein:

said distributed operation is a distributed database transaction being executed by a set of processes coordinated by a coordinator process;

the method further includes the step of said coordinator process causing a new process on a database server to participate in said distributed database transaction; and

the step of causing updates to said participant data includes said coordinator process causing updates to said participant data in response to said new process participating in said distributed database transaction.

6. (No Change) The method of Claim 1, wherein

said distributed operation is a distributed database transaction;

the step of registering includes registering participant data that identifies which database servers of a plurality of database servers are participating in said distributed database transaction; and

the step of causing a node to retrieve said participant data includes causing a node that requires information about participants in said distributed database transaction to retrieve said participant data from said name service.

7. (No Change) The method of Claim 1, wherein:

said distributed operation is a distributed database transaction;

the method further includes the step of assigning a transaction identifier to said distributed database transaction;

the step of registering includes registering in said name service data that associates said participant data with said transaction identifier; and

the step of causing a node includes causing a node to request from said name service published data associated with said transaction identifier.

8. (No Change) The method of Claim 1, wherein the step of causing a node to retrieve said participant data includes said name service process receiving a request from a first process to supply said participant data, wherein said name service process and said first process reside on said node.

9. (No Change) The method of Claim 8, wherein the step of causing a node to retrieve said participant data includes said name service process retrieving said participant data from one or more data structures residing on said node in response to receiving said request.

10. (No Change) The method of Claim 1, wherein the step of causing a node to retrieve said participant data includes a name service process receiving a request from a first process to supply said participant data, wherein said name service process and said first process reside on said node.

11. (Amended) A computer-readable medium carrying one or more sequences of one or more instructions for determining participants of a distributed operation in a distributed system, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

registering in a name service participant data that identifies a plurality of participants that are participating in said distributed operation; **[and]**

wherein said distributed operation is a unit of work involving said plurality of participants;

wherein said name service registers information received from clients and provides said information to clients that request the information; and

causing a node that requires information about participants in said distributed operation to retrieve said participant data from said name service.

12. (No Change) The computer-readable medium of Claim 11, wherein the step of causing a node to retrieve said participant data includes causing said node to retrieve said participant data in response to said node performing deadlock detection.

13. (No Change) The computer-readable medium of Claim 11, wherein:

said distributed operation is a distributed transaction; and

the step of registering includes registering in a name service participant data that identifies which database servers of a plurality of database servers are participating in said distributed transaction.

14. (No Change) The computer-readable medium of Claim 11, further including the step of causing updates to said participant data to identify a new participant in said distributed operation.

15. (No Change) The computer-readable medium of Claim 14, wherein:

said distributed operation is a distributed database transaction being executed by a set of processes coordinated by a coordinator process;

the computer-readable medium further includes sequences of instructions for performing the step of said coordinator process causing a new process on a database server to participate in said distributed database transaction; and

the step of causing updates to said participant data includes said coordinator process causing updates to said participant data in response to said new process participating in said distributed database transaction.

16. (No Change) The computer-readable medium of Claim 11, wherein

said distributed operation is a distributed database transaction;

the step of registering includes registering participant data that identifies which database servers of a plurality of database servers are participating in said distributed database transaction; and

the step of causing a node to retrieve said participant data includes causing a node that requires information about participants in said distributed database transaction to retrieve said participant data from said name service.

17. (No Change) The computer-readable medium of Claim 11, wherein:

said distributed operation is a distributed database transaction;

the computer-readable medium further includes sequences of instructions for performing the step of assigning a transaction identifier to said distributed database transaction;

the step of registering includes registering in said name service data that associates said participant data with said transaction identifier; and

the step of causing a node includes causing a node to request from said name service published data associated with said transaction identifier.

18. (No Change) The computer-readable medium of Claim 11, wherein the step of causing a node to retrieve said participant data includes said name service process receiving a request from a first process to supply said participant data, wherein said name service process and said first process reside on said node.

19. (No Change) The computer-readable medium of Claim 18, wherein the step of causing a node to retrieve said participant data includes said name service process retrieving said participant data from one or more data structures residing on said node in response to receiving said request.

20. (No Change) The computer-readable medium of Claim 1, wherein the step of causing a node to retrieve said participant data includes a name service process receiving a request from a first process to supply said participant data, wherein said name service process and said first process reside on said node.